## CLAIMS

1.     In a J2EE application server, a method for monitoring performance of a plurality of transactions including a top level transaction and plurality of transactions relating to said top level transaction in a child parent hierarchy, comprising

for each of selected ones of said transactions, instrumenting said transaction at run-time without modifying its source code to obtain a performance metric corresponding thereto,

for each of said instrumented transactions, generating a correlator for identifying said top level transaction and a parent, if any, of said transaction, and

utilizing said correlators to cross-correlate a performance metric corresponding to a parent transaction with one or more performance metrics corresponding to one or more child transactions of said parent transaction.

2.     The method of claim 1, wherein the step of instrumenting said transaction comprises inserting instrumentation code in a bytecode representation of said transaction.

3.     The method of claim 2, wherein said performance metric corresponds to a response time of said transaction.

4.     The method of claim 3, wherein said instrumentation code effects generation of a start time marker upon start of execution of said transaction and generation of a stop time marker upon completion of execution of said transaction.

5.     The method of claim 4, wherein said instrumentation code generates calls to an Application Response Measurement (ARM) agent to cause generation of said stop and start time markers.

6.     The method of claim 5, further comprising utilizing said start and stop time markers to measure a response time of said transaction.

7.    The method of claim 1, further comprising generating a record for each instrumented transaction upon completion of said transaction, said record indicating said performance metric associated with said transaction, a parent of said transaction, and said top level transaction.

5

8.    The method of claim 7, further comprising transmitting said transaction record to an analysis and presentation module.

9.    The method of claim 1, further comprising storing said correlators in a thread local

10   storage stack in case of execution of said hierarchical transactions in a single thread.

10.   The method of claim 9, further comprising storing said correlators in the stack based on a LIFO protocol.

15   11.   The method of claim 10, further comprising removing a correlator from said stack upon completion of a transaction associated with said correlator.

12.   The method of claim 1, wherein said top level transaction is initiated in response to a request received from a web server.

20

13.   The method of claim 12, wherein said web server transmits a cookie to said application server together with said request.

14.   The method of claim 13, further comprising

25          utilizing said cookie to generate said top level correlator.

15.     A method for monitoring performance of at least two Java transactions executing in
two separate processes and being related to one another as parent-child transactions,
comprising

5          instrumenting each of said transactions at run-time by modifying its respective
bytecode representation to obtain a selected performance metric corresponding thereto,

           generating a correlator corresponding to said top-level transaction,

10         utilizing RMI over IIOP to send said top-level correlator incorporated in a header of
an IIOP message to said child transaction, and

           generator another correlator corresponding to said child transaction.

15   16.   The method of claim 15, further comprising
           employing said correlators to cross correlate the performance metric of said top level
transaction with the performance metric of said child transaction.

     17.   The method of claim 15, wherein said performance metric corresponds to a response
20   time of said transaction.

     18.   The method of claim 17, wherein said modified bytecode representation effects
generation of a start time marker upon start of execution of said transaction and a stop time
marker upon completion of execution of said transaction.

25

     19.   The method of claim 18, wherein said modified bytecode representation generate calls
to an Application Response Measurement (ARM) agent to cause generation of said start and
stop time markers.

30

20.    A computer readable medium for storing instructions for instrumenting at run-time a hierarchical chain of parent-child transactions including a to top level transition and at least one child transaction thereof without modifying source codes associated with these transactions and for generating correlators for each of said transactions, wherein each correlator identifies said top level transation and a parent, if any, corresponding to its associated transaction.